Stock Price Prediction Using Univariate Time Series

Saurabh Bhise
Department Of Computing
Dublin City University
Dublin, Ireland
saurabh.bhise2@mail.dcu.ie
19210665

Nivedita Mahato
Department Of Computing
Dublin City University
Dublin, Ireland
nivedita.mahato2@mail.dcu.ie
19210398

Geha Mehta
Department Of Computing
Dublin City University
Dublin, Ireland
geha.mehta3@mail.dcu.ie
19210829

***Abstract*—This paper focuses on the study is to compare the performance of time series models on predicting the stock prices of different organizations. The analysis of the stock prices is an essential part of the finance industry. Time series prediction can be applied on any set of variables that change over a period. Statistical handling of time series data is called extrapolation, which is the prediction of the future. Traditionally, statistical methods such as Auto-Regressive Integrated Moving Average (ARIMA) demonstrated high performance in terms of accuracy in predicting future values. However, with the recent advancement of new machine learning algorithms, new algorithms are being developed to forecast time series. Long Short Term Memory (LSTM) is yet another neural network technique which is superior to traditional statistical methods. In this comparative study, the models are trained and tested on the data to obtain and check which model performed the best in term of accuracy. Historical stock price data of twelve years was obtained from Yahoo Finance to build these models. These models have performed well with the data and have shown great potential in predicting time series. The performance of these models has been evaluated based on the metrics such as Root Mean Squared Error (RMSE) and Mean Standard Percentage Error (MAPE).**

***Index Terms*—Stock price prediction,Linear Regression, ARIMA, LSTM**

I. Introduction

A. Background and Motivation

Stock markets are some of the most significant aspects of the global economy today. The economic growth of developing countries depends on stock markets. Any variations in the stock market impact the personal and corporate financial lives and the economic health of the country. It is the most popular form of investments because of its high returns. Also, there is always a risk associated with an investment in the stock market because of its highly unpredictable behaviour. Prediction of the stock market is a classic problem which has been extensively researched using machine learning methodologies [1]. Analysts in the financial sector are entrusted with the forecasting of stock market prices, exchange rates, loan default rates or stock market indices. A common motif in the problems as mentioned above revolves around the need to evaluate past or current variable (or variables) observations to anticipate future observations values. Because of the broad significance and multidisciplinary application of time series, there is an increase in research studies which are focused on developing the techniques for making accurate time series predictions [2].

The most widely used method for univariate time series prediction is ARIMA which is a combination of two models: Auto-Regressive (AR) and Moving Average (MA). For multivariate time series prediction, methods such as Vector Auto-Regression (VAR) and multivariate ARIMA are used which allow the use of more than one input variable. With the recent development of new machine learning algorithms and deep learning techniques, researchers are finding a new approach to predict the time series problems where the relationships are modelled in a complex hierarchy of variables. Machine learning model like Support Vector Machine (SVM) and deep learning methods such as Long Short Term Memory(LSTM) have gained high popularity in time series forecasting.

B. Research Question

"How to forecast stock prices of Apple Inc. using univariate historical financial time series by applying data mining techniques such as Linear Regression, Time Series Forecasting and Recurrent Neural Networks?"

The research question of this study will address which data mining technique has outperformed the other techniques by achieving higher accuracy in terms of prediction of future stock prices of Apple Inc using historical daily stock prices as an input variable.

C. Research Objective

The main objective of this study is to analyse and predict the stock prices of the Apple Company. For our study, we selected three different approaches, i.e. Statistical methods, Time Series Forecasting and Neural Networks. In each of the approaches, we have selected Linear Regression, Seasonal ARIMA and Long Short Term Memory (LSTM) machine learning algorithms respectively. We intend to analyse and compare each model and check which approach performed which high accuracy in terms of prediction of the stock price.

The following paper is organised as follows: Section II gives a brief literature review about the methods for time series analysis. Section III discusses the methodology that

has been undertaken in this research. Section IV describes the implementation process for the algorithms. The evaluation metrics of the analysis are presented in Section V. Section VI discusses the conclusion of this research study.

II. Literature Review

A. Statistical Methods

The theory and practice of regression techniques for predicting stock price patterns using a transformed dataset are transformed into ordinal data set. The initial pre-transformed database provides data on different data forms used for currency values and financial ratios [3]. Data formats for currency values and financial ratios provide a method for the measurement of stock prices. Regression models are built using the Waikato Environment for Knowledge Analysis (WEKA) machine learning software which analysed the stock price movement in Bursa Malaysia. In the study, Linear regression model outperformed the other regression models. The paper also suggests the use of a different data type by transforming real numbers into categorical, ordinal data that can improve the outcomes of the regression techniques.

Some work on sentiment analysis, [4] has been used to forecast stock prices. Sentiment analysis is a method to evaluate emotion inside people; it may help some businesses. As there was an increase in the use of social media in recent times, [4] in his research concludes that people opinion on social media such as Twitter can predict Dow Jones Industrial Average (DJIA) value with 87.6% accuracy. This shows that there is a relation between sentiment analysis and stock prices. The Linear regression model is used to predict the company stock price. Similarly, [5] Data analysis techniques are used to assist investors in making the correct financial forecasts which will help the investors to take the best decisions. The paper used linear regression and ANN to predict the stock value of different companies using the Nifty 50 stock price. The data was of 9 years. The accuracy was estimated using 2-3 years of forecast results for R and two months of forecast results for Python after comparison with the original stock price. The paper showed that Linear Regression has less error when a feature like the opening price of the stock is used in forecasting the stock price.

Stock prediction is researched in economics and data mining studies. Stock forecasts earned special attention due to their significance in creating more successful and productive planning. In the paper [6], they developed a mobile application which predicts the stock price of "Jakarta Composite Index (A JKSE)" fetched from Yahoo Finance. A Multiple Linear Regression (IMLR) was built with Moving Average technique. Time Series analysis helped in predicting the stock price. The mobile device accuracy forecast gives a better result than the standard algorithm with a value of 15087.465 in MSE, 122.831 in RMSE, and 3.255 in MAPE. These papers helped us in understanding that the linear regression model performs well with time-series datasets.

B. Time Series Forecasting

Several researchers have proposed different models for the analysis of time series in the past. ARIMA is one of the traditional statistical models which has proven to be effective in terms of time series prediction. In the research undertaken by [7], the study focused on building ARIMA models for short term stock price prediction. The models were applied on the historical data obtained from Nokia Stock Index and Zenith Bank Index. The closing price was selected as the input variable for the model. R Squared and Standard Error were used as the evaluation metrics in the research.

A study conducted by the researchers approached the problem of predicting stock prices by the use of another variant of statistical methods called Seasonal ARIMA [8]. The study aimed at providing a parameter estimation and the diagnostic checking procedures to the model. The data was extracted from the NASDAQ index, and the monthly average of 'Low' and 'High' prices was calculated and selected as input to the models. The study also focused on the comparison between the seasonal ARIMA model and the non-seasonal ARIMA model. This research [8] concluded that the Seasonal ARIMA model performed better than the non-seasonal ARIMA model.

Also, the researches [9] considered the decomposition benefits of wavelet transformation, which can decompose the time series data into several scales where both approximate and detailed part of the data can be obtained, proposed a hybrid forecasting scheme which combined the classic SARIMA method and wavelet transform (SW) and was successful for forecasting sales time series with a highly volatile pattern and in extracting a time series features which influence the forecasting accuracy. The model used three different datasets to compare its performance and accuracy with two other models, namely pure SARIMA and the Classical Seasonal Decomposition (CSD) with Linear Extrapolation with Seasonal Adjustment method (LESA). The comparison showed that wavelet method outperforms all other methods when the dataset exhibits highly volatile pattern and CSD+LESA model performs better than the others when the dataset holds strong white noise and SARIMA model outperforms the above two when seasonality in the dataset is extreme.

C. Neural Networks

The research in [10], which uses the stock price of China stock market in Shanghai and Shenzhen from Yahoo Finance, has been considered for the analysis. The LSTM model shows the different result of prediction when used with a different number of features and using a different combination of features. The paper also tries to explain how the different number of LSTM layers also affects the result of the model. In 2017, [11] used a sliding window approach along with the three different deep learning architectures, RNN, LSTM and CNN for short term Stock price prediction of a univariate series of NSE stock index. Three univariate series two from IT sector and one from pharma sector were selected for the sliding window approach where the window-size was fixed to 100 with an overlap of 90 minute's information and 10 minutes of

future prediction. Later this output was used with RNN, LSTM and CNN for prediction. The result showed that CNN provided more accurate results than the other two models because of the properties of not depending on any previous information and using the current window information for prediction.

In [12], exhibited the temporal characteristics of the stock market and the Long Short Term Memory (LSTM) recurrent neural network to analyse the stock data, extract feature value and help in the prediction model of the stock market prices. LSTM is a different type of neural network which can preserve its error along with layer and time. In this paper, the authors have used the characteristics of LSTM algorithm to predict short term changes of the corresponding stock transactions.

In [13], used a deep learning framework comprising bidirectional ConvLSTM stacked autoencoders for univariate time series prediction. The execution was carried in three phases, in the first phase Wavelet Decomposition/Transformation was used to denoise the time series data, in the second phase Stacked AutoEncoders (SAE), architecture, was used for feature extraction and sequence learning in an unsupervised manner and in the third phase a two-dimensional bidirectional ConvLSTM is used for accurately predicting time series. The approach was tested on three openly available univariate time series datasets, and the proposed model was also compared with two state-of-the-art prediction models i) CNN-LSTM Autoencoder (CNN-LSTM-SAE) and (ii) 2D Convolutional LSTM (2DConvLSTM).

III. Data Mining Methodology

A. Knowledge Discovery Databases

The Knowledge Discovery Databases (KDD) is an iterative model. It is an exploratory process of extracting hidden knowledge from the database. To further improve the discovered knowledge, this process undergoes several pre-processing and post-processing techniques. It requires a brief understanding of the application domain and goals.

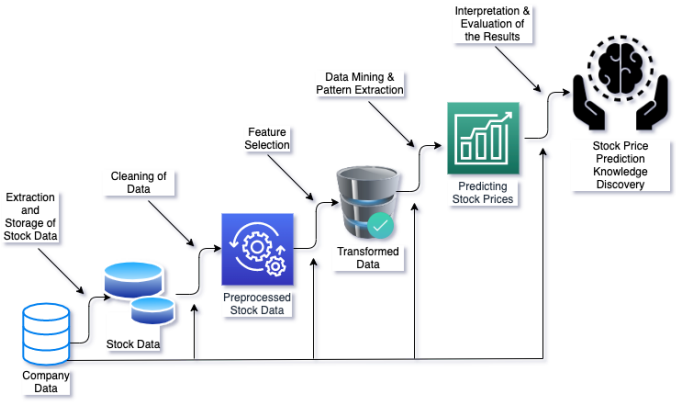

Fig. 1. KDD methodology for Stock Price Prediction

1) Data Selection: The financial time series data of the 'Apple' Company is extracted dynamically from 'Yahoo Finance' for the period from January 2008 to January 2020 and is composed of the seven variables, namely: 'Data', 'High', 'Low', 'Open', 'Close', 'Volume', 'Adjusted Close'. For this study, we have selected 'Adjusted Close' price as the input variable.

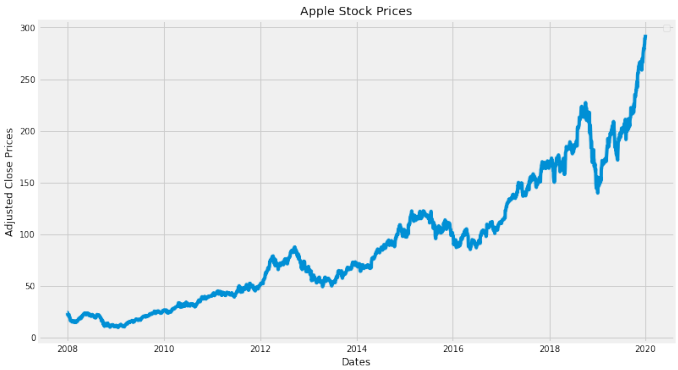

Fig. 2. Adjusted Close Price

2) Data Pre-processing: The appropriate pre-processing of data plays a vital role in determining the efficiency of the prediction model. In this step, it is necessary to keep the data as accurate as possible. Pre-processing steps involve the detection and corrections of the errors, checking for null values and filling in the missing values.

3) Data Transformation: The time-series data of Apple stocks were divided into two sets: training and testing dataset. The training dataset accounted for 80% whereas the test dataset accounted for 20 %. The models were applied to the training dataset, and the prediction was made on the test dataset. Then the predicted values were compared with the values in the test dataset for the comparison of the models.

Feature scaling is used to normalize the number of independent variables or data features. We have used Rescaling (min-max normalization) in our dataset. Rescaling is the most straightforward approach and consists of rescaling the number of features to be scaled to [0, 1] or [-1, 1]. The target range is decided on the nature of the data. For our dataset, we have selected the range [0, 1].

4) Data Mining: In this phase, we try to find a function which models the transformed data with least error. Time series modelling will be implemented on LR, ARIMA and LSTM methods on the training dataset and the predictions will be carried out on test dataset.

5) Knowledge Discovery: After the models are applied to the data, the results will be evaluated using the evaluation methods. The details of the evaluation methods are described in section 5. The knowledge discovered from these findings will be visualised, and the report will be made available.

IV. Proposed Implementation

A. Linear Regression

The most common machine learning algorithm that can be applied to such data is linear regression. The model returns the equation that defines the correlation between the independent variables and the dependent variables. The equation for linear regression can be written as:

$$Y = \theta 1X1 + \theta 2X2 + \ldots + \theta nXn \quad (1)$$

Here, X1, X2...Xn represent the independent variables while the coefficients $\theta 1, \theta 2, \ldots, \theta n$ represent the weights. Our data includes only one independent variable, which is the date and the dependent variable. We are trying to forecast the stock price. The objective is to find these coefficients and to ensure that the Sum of Squared Errors, which represents the difference between each point in the dataset and the corresponding predicted value produced by the model, is minimal. We created a subset using the feature daily adjusted closing price 'Adj Close' as the value to predict in our model. The subset data frame has two columns 'Adj Close' contains numerical data and 'Date' is the index column and contains date-time values. To evaluate the linear regression model, we are interpreting the coefficients:

- The slope coefficient tells us that with a one-unit increase in date, the adjusted closing price increases by 0.0744 Dollars.
- The intercept coefficient is the price at which the closing price measurement started is 10.161 Dollars.

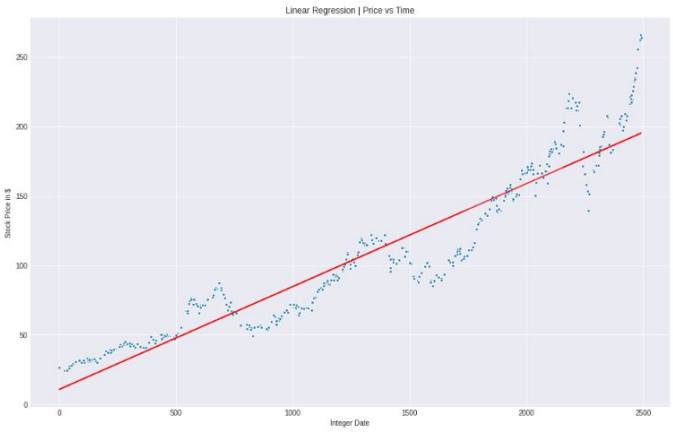

Fig. 3. Prediction using Linear Regression

The data points are mostly close to a diagonal showing us, that the predicted values are close to the actual value making the model quite good. However, there are some areas, around 100 to 120, the model seems to be quite random and shows no relationship between the predicted and actual value. Also, the predictions do not cover the values above 200. A sample dataset of 25 randomly selected values was created to see the predicted value of our model and the actual Adj Close values.

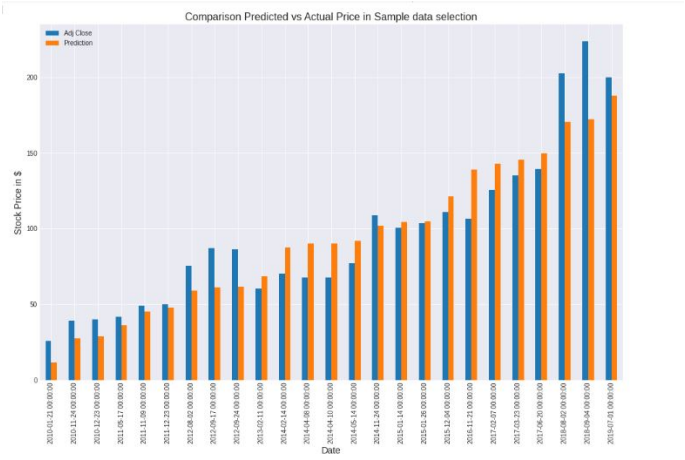

Fig. 4. Linear Regression Prediction on 25 Sample data

Error in the model is evaluated using different metrics as below:

```
Mean Absolute Percentage Error: 17.74686079243013
Root Mean Squared Error: 19.013428352343553
```

Fig. 5. Evaluation Result of Linear Regression

To see how accurate our model is, we may measure the Coefficient of Determination, which defines the ratio of the total error to the error, as described in our model. The value is between 0 and 1, with 0 indicating that the model fails to model the data accurately. The value of $R\hat{2}$ in our linear regression model is nearly 88.8%.

B. Seasonal Auto Regressive Integrated Moving Average

Seasonal ARIMA algorithm is implemented in Python. This model needs three parameters: ARIMA(p,d,q). The AR term is defined by p and the MA term by q whose values are determined by PACF and ACF plot, respectively. The term I, defined by d, is the order of differencing which specifies the number of times the series has been differentiated to make it stationary. We have used the Augmented Dickey-Fuller (ADF) test to check the stationarity of the series.

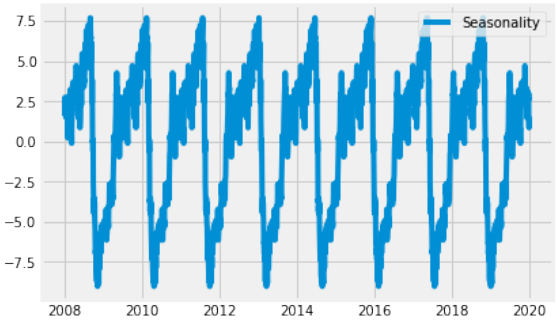

Fig. 6. Seasonality in time series

As we can see from the above figure that, there is seasonality in our series, which means that the series is not stationary.

To make the time series stationary, we have to tune the parameters of the model such that it passes the ADF test. We need to try various combinations of the p,d and q values to obtain stationarity. Also, we need to determine the combination of p,d and q, which yields the lowest Akaike's Information Criterion (AIC). Lower the value of AIC better the combination of the parameters.

```
Performing stepwise search to minimize aic
Fit ARIMA: (2, 2, 1)x(0, 0, 1, 12) (constant=True); AIC=11765.051, BIC=11801.127, Time=14.276 seconds
Fit ARIMA: (0, 2, 0)x(0, 0, 0, 12) (constant=True); AIC=13822.337, BIC=13834.362, Time=0.081 seconds
Fit ARIMA: (1, 2, 0)x(1, 0, 0, 12) (constant=True); AIC=13073.106, BIC=13097.157, Time=1.909 seconds
Fit ARIMA: (0, 2, 1)x(0, 0, 1, 12) (constant=True); AIC=11768.937, BIC=11792.988, Time=9.590 seconds
Near non-invertible roots for order (0, 2, 1)(0, 0, 1, 12); setting score to inf (at least one inverse root too close to the border of the unit circle: 1.000)
Fit ARIMA: (0, 2, 0)x(0, 0, 0, 12) (constant=False); AIC=13820.337, BIC=13826.350, Time=0.065 seconds
Fit ARIMA: (2, 2, 1)x(0, 0, 0, 12) (constant=True); AIC=11763.855, BIC=11793.918, Time=4.748 seconds
Near non-invertible roots for order (2, 2, 1)(0, 0, 0, 12); setting score to inf (at least one inverse root too close to the border of the unit circle: 1.000)
Fit ARIMA: (2, 2, 1)x(1, 0, 1, 12) (constant=True); AIC=11767.069, BIC=11809.158, Time=18.490 seconds
Near non-invertible roots for order (2, 2, 1)(1, 0, 1, 12); setting score to inf (at least one inverse root too close to the border of the unit circle: 0.999)
Fit ARIMA: (2, 2, 1)x(0, 0, 2, 12) (constant=True); AIC=11766.414, BIC=11808.503, Time=68.186 seconds
Near non-invertible roots for order (2, 2, 1)(0, 0, 2, 12); setting score to inf (at least one inverse root too close to the border of the unit circle: 0.997)
Fit ARIMA: (2, 2, 1)x(1, 0, 0, 12) (constant=True); AIC=11765.495, BIC=11801.571, Time=17.168 seconds
Near non-invertible roots for order (2, 2, 1)(1, 0, 0, 12); setting score to inf (at least one inverse root too close to the border of the unit circle: 0.998)
Fit ARIMA: (2, 2, 1)x(1, 0, 2, 12) (constant=True); AIC=11766.599, BIC=11814.701, Time=64.717 seconds
Near non-invertible roots for order (2, 2, 1)(1, 0, 2, 12); setting score to inf (at least one inverse root too close to the border of the unit circle: 0.999)
Fit ARIMA: (1, 2, 1)x(0, 0, 1, 12) (constant=True); AIC=11770.642, BIC=11800.706, Time=12.766 seconds
Near non-invertible roots for order (1, 2, 1)(0, 0, 1, 12); setting score to inf (at least one inverse root too close to the border of the unit circle: 1.000)
Fit ARIMA: (2, 2, 0)x(0, 0, 1, 12) (constant=True); AIC=12706.019, BIC=12736.083, Time=2.243 seconds
Fit ARIMA: (3, 2, 1)x(0, 0, 1, 12) (constant=True); AIC=11765.600, BIC=11807.689, Time=26.314 seconds
Near non-invertible roots for order (3, 2, 1)(0, 0, 1, 12); setting score to inf (at least one inverse root too close to the border of the unit circle: 0.999)
Fit ARIMA: (2, 2, 2)x(0, 0, 1, 12) (constant=True); AIC=11775.029, BIC=11817.117, Time=24.806 seconds
Near non-invertible roots for order (2, 2, 2)(0, 0, 1, 12); setting score to inf (at least one inverse root too close to the border of the unit circle: 1.000)
Fit ARIMA: (1, 2, 0)x(0, 0, 1, 12) (constant=True); AIC=13073.105, BIC=13097.156, Time=1.850 seconds
Fit ARIMA: (1, 2, 2)x(0, 0, 1, 12) (constant=True); AIC=11771.562, BIC=11807.638, Time=28.063 seconds
Near non-invertible roots for order (1, 2, 2)(0, 0, 1, 12); setting score to inf (at least one inverse root too close to the border of the unit circle: 0.999)
Fit ARIMA: (3, 2, 0)x(0, 0, 1, 12) (constant=True); AIC=12436.551, BIC=12472.627, Time=2.574 seconds
Fit ARIMA: (3, 2, 2)x(0, 0, 1, 12) (constant=True); AIC=11768.027, BIC=11816.928, Time=25.442 seconds
Near non-invertible roots for order (3, 2, 2)(0, 0, 1, 12); setting score to inf (at least one inverse root too close to the border of the unit circle: 0.999)
Total fit time: 325.737 seconds
11763.854927882356
```

Fig. 7. Selection of p,d,q parameters with the least value of AIC

Even Though ARIMA is a compelling model; the parameter tuning process becomes time-consuming. In Auto ARIMA, we only need to specify the range of values for p, d and q and our data being seasonal as observed from the visualisation, we need to set the seasonal parameter to be True. The model automatically calculates the lowest value of AIC through multiple combinations of the p,d and q parameters and returns the final values of the parameters. This can be observed in Fig. 6.

Following figure shows the prediction of our model against the test data.

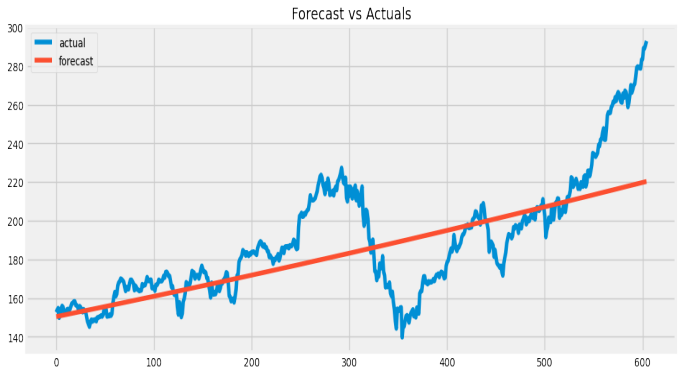

Fig. 8. Prediction using Seasonal ARIMA

The results are calculated using the MAPE and RMSE metrics.

```
{'Correlation': 0.7437013557505916,
 'MAPE': 0.08034745842315415,
 'RMSE': 21.887124728811614}
```

Fig. 9. Evaluation Result of Seasonal ARIMA

C. Long Short Term Memory

Long Short-Term Memory (LSTM) is used for sequential prediction and has been proven effective. The reason they perform so well is that LSTM is capable of processing past information that is important and losing information that is not important. LSTM has three gates:

- The input gate: It adds the information to the cell state.
- The forgot gate: It removes information that is no longer needed by the model.
- The output gate: Output gate at the LSTM selects the information to be displayed as output.

In LSTM model, data scaling was done to transform the data so that it fits within a specific scale, for our model we used the scale 0–1 which is part of the feature scaling done to normalize the data within a particular range. A 2-layer Sequential LSTM Model was built for predicting and forecasting the stock price.

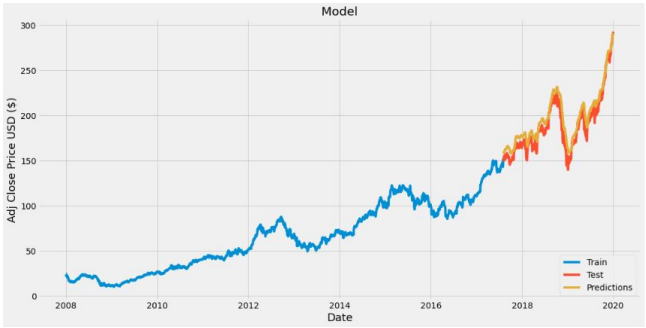

Fig. 10. Prediction using LSTM

The error in the model is evaluated using different metrics as below:

```
Mean Absolute Percentage Error: 2.310485255280895
Root Mean Squared Error: 5.791423706290812
```

Fig. 11. Evaluation Result of LSTM

V. Proposed Evaluation Methods

A. Root Mean Squared Error

Root Mean Squared Error (RMSE) can be thought of as normalised distance between the vector of predicted values and that of observed values. It is a good measure to make use of if we want to estimate the SD of an observed value from our model's prediction. If RMSE is large, this generally

means our model is failing to account for essential features underlying our data. It represents the standard deviation of the residuals, i.e. the difference between the model predictions and the original (train data) RMSE unit matches the units of the output. It gives an approximation of the scale of the residuals that are being dispersed.

It is given by:

$$RMSE = \sqrt{\Sigma_{i=1}^{n} \frac{(p_i - o_i)}{n}^2}$$

Where,

- p1...pn are predicted values
- o1...on are observed values
- n is the number of observations

B. Mean Absolute Percentage Error

Mean Absolute Percentage Error (MAPE) is a measure of the accuracy of the forecasting system. This accuracy is measured as a percentage and can be calculated as the absolute average per cent error for each period minus the actual values divided by the actual values. Works best if there are no extremes to the data and no zeros. Its range can vary from 0 to infinity which makes it challenging to interpret the result as compared to the training data. MAPE exhibits some limitations if the data point value is zero.

It is given by:

$$MAPE = \frac{100\%}{n} \Sigma_{i=1}^{n} | \frac{(y - yi)}{yi} |$$

Where,

- y...yn are observed values
- yi...yin are predicted values
- n is the number of observations

TABLE I
EVALUATION RESULTS

Methods	*MAPE*	*RMSE*
Linear Regression	17.74	19.01
ARIMA	8.03	21.88
LSTM	2.31	5.79

VI. CONCLUSIONS AND FUTURE WORK

The objective of this study was to predict the future stock prices of Apple Inc using univariate time series forecasting. This analysis compares the accuracy of Linear Regression, Seasonal ARIMA and LSTM, as representative techniques when forecasting time series data. These techniques have been implemented and applied on historical stock price data, and the results have been evaluated with the help of evaluation metrics such as Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE). The data was dynamically collected from Yahoo Finance as it provides up to date data without any missing values. The result of the analysis showed that the LSTM technique performed significantly better than the other two techniques.

The analysis discussed in this paper promotes the benefits of applying deep learning algorithms and techniques for financial time series data. In future, to further improve the accuracy of these models, more than one variable (multivariate analysis) can be considered.

REFERENCES

[1] P. Somani, S. Talele and S. Sawant, "Stock market prediction using Hidden Markov Model," 2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference, Chongqing, 2014, pp. 89-92.

[2] A. Essien and C. Giannetti, "A Deep Learning Framework for Univariate Time Series Prediction Using Convolutional LSTM Stacked Autoencoders," 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), Sofia, Bulgaria, 2019, pp. 1-6.

[3] H. L. Siew and M. J. Nordin, "Regression techniques for the prediction of stock price trend," 2012 International Conference on Statistics in Science, Business and Engineering (ICSSBE), Langkawi, 2012, pp. 1-5.

[4] Y. E. Cakra and B. Distiawan Trisedya, "Stock price prediction using linear regression based on sentiment analysis," 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS), Depok, 2015, pp. 147-154.

[5] S. Tiwari, A. Bharadwaj and S. Gupta, "Stock price prediction using data analytics," 2017 International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, 2017, pp. 1-5.

[6] A. Izzah, Y. A. Sari, R. Widyastuti and T. A. Cinderatama, "Mobile app for stock prediction using Improved Multiple Linear Regression," 2017 International Conference on Sustainable Information Engineering and Technology (SIET), Malang, 2017, pp. 150-154.

[7] A. A. Ariyo, A. O. Adewumi and C. K. Ayo, "Stock Price Prediction Using the ARIMA Model," 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, 2014, pp. 106-112.

[8] W. Wang and Z. Niu, "Time Series Analysis of NASDAQ Composite Based on Seasonal ARIMA Model," 2009 International Conference on Management and Service Science, Wuhan, 2009, pp. 1-4.

[9] Choi, T.M., Yu, Y. and Au, K.F., 2011. A hybrid SARIMA wavelet transform method for sales forecasting. Decision Support Systems, 51(1), pp.130-140.

[10] K. Chen, Y. Zhou and F. Dai, "A LSTM-based method for stock returns prediction: A case study of China stock market," 2015 IEEE International Conference on Big Data (Big Data), Santa Clara, CA, 2015, pp. 2823-2824.

[11] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon and K. P. Soman, "Stock price prediction using LSTM, RNN and CNN-sliding window model," 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, 2017, pp. 1643-1647.

[12] S. Liu, G. Liao and Y. Ding, "Stock transaction prediction modelling and analysis based on LSTM," 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), Wuhan, 2018, pp. 2787-2790.

[13] [7.6] A. Essien and C. Giannetti, "A Deep Learning Framework for Univariate Time Series Prediction Using Convolutional LSTM Stacked Autoencoders," 2019 IEEE International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), Sofia, Bulgaria, 2019, pp. 1-6.